REMARKS

In the Office Action mailed May 6, 2004, amendments to the specification and drawings were either suggested or required. Applicants have so amended the specification and drawings and added new Claims 17-26.

Claims 1-16 were further rejected for "same invention" double patenting over Claims 1-16 of co-pending Applications Nos. 10/164,789 and 10/165,457. Applicants respectfully submit that this double patenting rejection is moot since preliminary amendments canceling Claims 1-16 were filed in each of these co-pending applications.

Finally, all of the claims 1-16 were rejected over, inter alia, the Walters et al and Davidson et al patents. Applicants respectfully traverse these rejections and submit that the claims at issue are patentable over those patents for the following reasons.

INDEPENDENT CLAIM 13

Applicants respectfully traverse the rejection of Claim 13 as anticipated by the Walters et al '593 patent. According to a pertinent aspect of Applicants' disclosure to which Claim 13 is directed, translation of a given portion of program code instructions may, for example, yield two different sets of target code instructions, depending upon the particular set of entry conditions which exist at the time that the given portion of program code is encountered. One may refer to the two sets of entry conditions as the "prevailing set" and the "subsequent set" of conditions. The "subsequent set" of conditions will be synonymously referred to hereafter as the "second set" of conditions.

According to one aspect of Applicants' method of generating target code, upon first encountering the given portion of program code, only the target code necessary to execute that portion of program code under the "prevailing set" of entry conditions is initially generated and stored. When the same portion of program code is again encountered, a determination is made as to whether target code corresponding to the "second set" of conditions ("the subsequent conditions") has been generated. If not, the target code necessary to execute the same portion of program code under that second set of conditions is generated.

Referring to the Walters et al. patent, and particularly to column 7 cited by the Office Action, there is no disclosure of the concept of generating different sets of target code for the same portion of program code (e.g., the same sequence of program code instructions), nor is there any such disclosure in Davidson et al. Since Claim 13 clearly recites generating different target code for, inter alia, the same "given portion of program code," that Claim is clearly not anticipated by Walters et al. Accordingly, Applicants respectfully submit that the Section 102 rejection of Claim 13 should be withdrawn and the Claim allowed.

CLAIMS 1-10 AND 15

Independent Claim 1 recites, inter alia, generating different intermediate representations for the same "given portion" of program code in response to respective "previous" and "subsequent" conditions. Again, neither Walters et al. nor Davidson et al. disclose or suggest such an intermediate representation generation method. Accordingly, combining those references fails to create even a prima facie case of obviousness. Accordingly, Applicants respectfully submit that the rejection of Claim 1 and its independent Claims 2-10 should be withdrawn.

The Office Action also sets forth a rejection of independent Claim 15 based on the same grounds of rejection as Claim 1. The reasons demonstrating allowability advanced for Claim 1 accordingly apply to Claim 15, which should therefore also be allowed.

CLAIMS 11-12 AND 16

Applicants further respectfully submit that the rejection of independent Claim 11 as obvious in view of Davidson et al and Walters et al. is also not well-founded.

In particular, the Office Action relies on the assertion that Davidson et al disclose the feature of Claim 11 where "intermediate representation is only initially generated and stored as is required to execute that block of program code with a then prevailing set of conditions," citing Col. 7, line 66 to Col. 8, line 17 of Davidson et al. The cited portion of Davidson et al reads as follows:

> The front end 20 parses the source code to identify tuples, then to identify basic blocks of code. A block of code is defined to be a sequence of tuples with no entry or exit between the first and last tuple. Usually a block starts with a label or routine entry and ends with a branch to another label. A task of the front end 20 is to parse the source code 21 and identify the tuples and

> blocks, which of course requires the front end to be language specific. The tuple thus contains fields 41 that say whether or not this tuple is the beginning of a block, and the end of a block
>
> As discussed in more detail below, one feature of the invention is a method of representing effects. A tuple has effects if it stores or writes to a memory location (represented at the IL level as a symbol), or is dependent upon what another tuple writes to a location. Thus, in the example given in FIG. 3, tuple $4 has an effect (store to I) and tuple $1 has a dependency (content of J). Thus the tuple data structure as represented in FIG. 4 has fields 42 and 43 to store the effects and dependencies of this tuple.

Applicant respectfully submits that the foregoing excerpt of Davidson et al. plainly fails to constitute a disclosure of generating only that intermediate representation which is necessary to execute a block of program code with a then prevailing set of conditions.

Therefore, Applicants respectfully submit that the rejection of independent Claim 11 and Claim 12 dependent therefrom should be withdrawn, and that those Claims should be allowed. Since Claim 16 was rejected on the same basis as Claim 11, the rejection of Claim 16 should also be withdrawn.

INDEPENDENT CLAIM 14

Applicants' independent Claim 14 recites a method of dynamically translating program code wherein the recited steps (a) - (c) are repeated in real time. Steps (a) – (c) include the steps of generating an intermediate representation of a block of first program code and then generating a block of second program code from the intermediate representation.

The Office Action proposes to meet the terms of Claim 14 by combining the teachings of Walters et al. and Davidson et al. In particular, the Office Action asserts that one skilled in the art would have been motivated to generate an intermediate representation as taught by Davidson et al. during the run-time translation process of Walters et al. The asserted basis for such motivation is "for the purpose of optimizing the program code regardless of the programming language." Applicants respectfully submit that the evidence of record demonstrates that there would be no motivation to combine the references and that the motivation proposed by the Office Action is not supported by the record.

First, it will be appreciated that Davidson et al. disclose what is known as a "static compiler." Such a compiler completely translates a particular program into another language before execution. The amount of time taken to perform such a translation is therefore relatively

unimportant when compared to methods of dynamic translation such as that addressed by Applicant's Claim 14.

On the other hand, the Walters et al '593 patent does relate to a dynamic run-time translator. Walters et al. generally discuss prior art interpreters which ran 10-20 times slower than native speed, and the Walters et al. disclosure itself is generally directed to methods for speeding up run-time execution. Therefore, one skilled in the art would clearly not be motivated to add an intermediate representation generation ("IR") layer into the Walters et al translator because of the prospect that the added complexity and overhead would tremendously slow down the Walters et al. cross-compilation system. The possibility of "optimization" would further not constitute a motivating factor because Walters et al. discloses that optimization is achieved within the context of his own non-IR based system (See e.g. Col. 3, lines 1-9).

Since the record reflects no motivation to combine the disparate systems of Walters et al and Davidson et al, Applicants respectfully submit that a prima facie case of obviousness has not been made out, and respectfully solicits allowance of Claim 14.

In view of the foregoing remarks and amendments, Applicants respectfully submit that the subject application is in condition for allowance. Applicants, therefore, respectfully request reconsideration and early notice of allowance.

Respectfully submitted,

PAUL, HASTINGS, JANOFSKY & WALKER LLP

Dated: 8-30-04

By: _____

Franklin D. Ubell
Reg. No. 27,009

PAUL, HASTINGS, JANOFSKY & WALKER LLP
P.O. Box 919092
San Diego, CA 92191-9092
Phone: (858) 720-2500
Fax: (858) 720-2555